

CISM exercise I: build model

From Interactive System for Ice sheet Simulation

Building CISM

The first step is to acquire the model code. In common with many open-source projects, there are regular *releases* of CISM — these are numbered snapshots of the code and represent stable versions of the model (relative to development versions of the code on the repository). The latest release can be downloaded from the CISM website^[1]:

<http://developer.berlios.de/projects/glimmer-cism>

Because we are currently working on readying Glimmer-CISM-2.0 for release, the most recent publicly available version on the code does not contain many recent updates that we would like to take advantage of in this course. For that reason, we will be working from an intermediate version of the code. To copy the tarball of that code into your working directory and unpack it type

```
cp /usr/projects/cesm/cism/CISM-LANL-4-2011.tar.gz ./
```

```
tar -xzvf CISM-LANL-4-2011.tar.gz
```

We then need to load the necessary modules and set some environmental variables, which we will do by running a script,

```
source /usr/projects/cesm/cism/cism-env-csh
```

Now, cd into the top level directory from the unpacked tarball,

```
cd CISM-LANL-4-2011
```

CISM uses autotools to build the code. First, we need to build the build system by typing

```
./bootstrap
```

Next, we need to run a configure script which, along with environmental variables we specified above, tells the code where to look to find the necessary NetCDF libraries, compilers, etc:

```
./configure
```

Once the configure step has finished, check to make sure that it was successful with,

```
tail config.log
```

If you see **configure: exit 0** as the final line then configure was successful. Now, we build the code by typing,

```
make
```

The code will take a while to build. When it is done, and assuming you haven't seen any obvious errors, you can confirm a successful build by checking in the *example-drivers/simple_glide/src/* directory. If there is an executable file there called **simple_glide**, then your build was successful. This is the executable we will use to run the various test cases using the higher-order dynamics version of the model.

Go to the second set of exercises.

Go to the third set of exercises.

Return to main coarse page

Retrieved from "http://websrv.cs.umd.edu/isis/index.php/CISM_exercise_I:_build_model"

- This page was last modified on 21 April 2011, at 13:36.